Optimization Based Rate Control for Multicast with Network Coding

Lijun Chen[†], Tracey Ho[†], Steven H. Low[†], Mung Chiang[‡] and John C. Doyle[†]

[†]Engineering & Applied Science Division, California Institute of Technology, USA

[‡]Department of Electrical Engineering, Princeton University, USA

Abstract—Recent advances in network coding have shown great potential for efficient information multicasting in communication networks, in terms of both network throughput and network management. In this paper, we address the problem of rate control at end-systems for network coding based multicast flows. We develop two adaptive rate control algorithms for the networks with given coding subgraphs and without given coding subgraphs, respectively. With random network coding, both algorithms can be implemented in a distributed manner, and work at transport layer to adjust source rates and at network layer to carry out network coding. We prove that the proposed algorithms converge to the globally optimal solutions for intra-session network coding. Some related issues are discussed, and numerical examples are provided to complement our theoretical analysis.

Index Terms—Rate control, Network coding, Multicast, Coding subgraph, Distributed algorithm.

I. INTRODUCTION

Network coding extends the functionality of network nodes from storing/forwarding packets to performing algebraic operations on received data. Starting with the work of [1], which shows that employing coding at intermediate nodes is sometimes needed to maximize multicast throughput, various potential benefits of network coding have been shown, including robustness to link/node failures [13] and packet losses [4], [18]. Distributed random linear coding schemes, see, e.g., [7], [3], have made practical implementation of network coding possible. In this paper, we address the problem of rate control at end-systems for network coding based multicast flows with elastic rate demand.

Most existing work on network coding considers coding among packets of each multicast session, and assumes that the communication rates for each session and the network link capacities are fixed and known. Given a cost function in terms of the flow on each link, a min-cost flow optimization problem is obtained and solved to find the optimal coding subgraphs, which specify how much of each session's data should be sent on each link, see, e.g., [19], [27], [30]. For this reason, we call coding subgraphs of this kind *capacitated* subgraphs.

However, in many practical networks, traffic is bursty and elastic with varying rates, and since the network is shared by many users with unknown or changing demands, the available link capacities are unknown and variable. In such cases, it is not practical to solve a min-cost flow optimization to obtain capacitated subgraphs. Also, rate control is needed to make full use of bandwidth while avoiding congestion and maintaining certain fairness among the competing flows in the network.

One approach we propose is to use coding subgraphs that are *un-capacitated* (i.e., specifying which links are used by a session but not the amount of the data sent on each link) and chosen based on general cost criteria that are independent of flow rates. This is a practical approach; most existing routing approaches specify analogously un-capacitated routes. Since each session uses only a limited set of trees, this approach may give lower rates compared to optimizing over the entire network, but it is much less complex. We give an algorithm that combines rate control at fast timescales and adaptive traffic shifting at slower timescales based on end-to-end congestion feedback in the network.

Another approach we consider does not explicitly find coding subgraphs, but makes dynamic routing and coding decisions based on queue length gradients. This approach, termed *back-pressure*, was first proposed for optimal routing and scheduling in [25] and extended to various contexts (e.g., [21], [16]) including network coding in [10]; our contribution in this part of the paper is to incorporate rate control with network coding.

Our consideration of rate control uses the framework of utility maximization, which can provide the flexibility of modelling user application needs or performance objectives and guide the design of distributed algorithms. As shown in, e.g., [12], [17], [14], TCP congestion control algorithms can be interpreted as distributed primal-dual algorithms over the Internet to maximize aggregate utility. We extend the basic utility maximization formulation to incorporate the two network coding approaches described above, and propose two corresponding dual-based adaptive rate control algorithms to meet the new challenges associated with network coding. With random network coding, both algorithms can be implemented in a distributed manner, and work at the transport layer to adjust source rates and at the network layer to do network coding. We prove that the proposed algorithms converge to the globally optimal solutions for intra-session network coding.

The main contribution of this paper is to present optimization models and propose adaptive, distributed rate control algorithms for network coding based multicast flows. Our algorithms are promising in practical implementation, and can be extended to handle different environments such as multilayer network coding and multirate multicasting.

II. RELATED WORK

There are several recent works on rate control of multicast flows, see, e.g., [11], [5], which consider traditional routing-based multicasting. In contrast, this paper studies rate control for network coding based multicasting.

With network coding, the work that are most similar to our work are [19], [30], [28], [29]. We use a similar model

but without network cost for the networks without given coding subgraphs, see subsection III-C. What differentiates our work from others are the following. First, we use a different decomposition and obtain a dynamic scheme that uses only local information, see section V. As an important consequence of such alternative decomposition, our solution requires the least communication overhead. Our solution can also be readily extended to the case with network cost. Second, our rate control algorithm is a dual subgradient algorithm whose dual variables admit concrete and meaningful interpretation as congestion prices. Third, our work also differs from [19], [30] in that we do not relax the network constraint (3) but to exploit it to do session scheduling.

The session scheduling component of our algorithm for the network without given coding subgraphs is similar to the joint network coding and scheduling algorithms in [10] for wired networks. However, our work is different in that we also includes the transport layer, and as such, the network will use congestion control to automatically explore the achievable rate region while optimizing some global objective for the end users.

All existing work on network coding solves for the optimal coding subgraphs based on a flow model that is similar to multicommodity flow model for routing [6]. However, as discussed in the Introduction, it is often impractical to do so. In analogy to what happens with routing, we consider the case where subgraphs are chosen based on general cost criteria. Thus, we also study rate control for networks with given coding subgraphs, see sections III-B and IV.

Related work also includes [22] that studies congestion control with adaptive multipath routing using a multi-commodity model for the routing. Our model for networks without given coding subgraphs is also a multi-commodity model but with the additional constraints from network coding, and moreover, we propose a different solution approach. For the case with given coding subgraphs, we use a technique similar to that from [22], [6].

III. MODELS AND PROBLEM FORMULATIONS

A. Network and Coding Model

Consider a network, denoted by a graph $\mathcal{G} = (N, L)$, with a set N of nodes and a set L of directed links. We denote a link either by a single index l or by the directed pair (i, j) of nodes it connects. Each link l has a fixed finite capacity c_l packets per second.

Let M denote the set of multicast sessions, indexed by m. Each session m has one source $s_m \in N^1$ and a set $D_m \subset N$ of destinations. Network coding allows flows for different destinations of a multicast session to share network capacity by being coded together: for a single multicast session m of rate x^m, information must flow at rate x^m to each destination; with coding the actual physical flow on each link need only be the maximum of the individual destination's flows [1]. These

---

[1]Our analysis can extend to handle multi-source multicasting in a straight-forward way.

---

constraints can be expressed as

$$
\sum_{j:(i,j)\in L} g_{i,j}^{md} - \sum_{j:(j,i)\in L} g_{j,i}^{md} = \begin{cases} x^m & \text{if } i = s_m \\ -x^m & \text{if } i = d \\ 0 & \text{otherwise} \end{cases} \quad (1)
$$
$$
\forall\, d \in D_m,
$$
$$
g_{i,j}^{md} \le f_{i,j}^m, \quad \forall\, d \in D_m, \quad (2)
$$

where for each link (i, j), $g_{i,j}^{md}$ gives the *information flow* for destination d of session m, and $f_{i,j}^m$ gives the *physical flow* for session m. Note that the information flow balance equation (1) is formally similar to the physical flow balance equation for routing of data flows in the network. The inequality (2) reflects the network coding condition relating physical rate and information rate:

$$
f_{i,j}^m = \max_d \{ g_{i,j}^{md} \}, \quad d \in D_m. \quad (3)
$$

Figure 1 gives an example, adapted from [1], of a linear network code, and the corresponding flow variables $(f_{i,j}, g_{i,j}^{d_1}, g_{i,j}^{d_2})$. For packet networks, the result is stated formally in Theorem 1 of [19], which we reproduce here, slightly adapted:

Theorem 1: The rate vector f satisfies the constraints (1)-(2) if and only if there exists a network code that sets up a multicast connection at rate arbitrarily close to x^m from source s_m to destinations in set D_m and that injects packets at rate arbitrarily close to $f_{i,j}$ on each link (i, j).
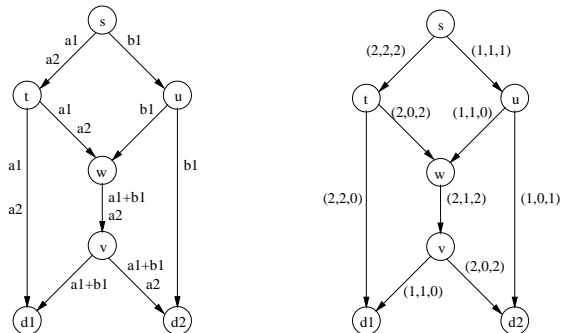

Fig. 1.   An example network coding subgraph with one source s and two destinations d_1 and d_2 (left graph), where links (s, u), (u, w) and (u, d_2) are assumed to have one unit of capacity, and all other links have two units of capacity; and the corresponding flow variables (right graph), where each link (i, j) is marked by the triple $(f_{i,j}, g_{i,j}^{d_1}, g_{i,j}^{d_2})$.

For the case of multiple sessions sharing a network, achieving optimal throughput requires in some cases coding across sessions. However, designing such codes is a complex and largely open problem. Thus, we limit our consideration to separate network codes operating within each session, an approach referred to as superposition coding [31] or intra-session coding. In this case, the set of feasible flow vectors is specified by combining constraints (1)-(2) for each session $m \in M$ with the following link capacity constraints:

$$
\sum_{m \in M} f_{i,j}^m \le c_{i,j}, \quad \forall\, (i, j) \in L. \quad (4)
$$

In practice, the network codes can be designed using the approach of distributed random linear network coding, see,

e.g., [7], [3], in which network nodes form output packets by taking random linear combinations of corresponding blocks of bits in input packets. The linear combination corresponding to each packet can be specified by a coefficient vector in the packet header, updated by applying to the coefficient vectors the same linear transformations as to the data. If (1)-(2) holds, each sink receives with high probability a set of packets with linearly independent coefficient vectors, allowing it to decode. The relative overhead of these coefficient vectors depends on parameters of the network code that can be chosen to trade-off overhead against performance, and it decreases with the size of the packets. See, e.g., [3], [10] for a detailed description and discussion of overhead and other practical implementation issues.

B. Multicast with Given Coding Subgraphs

We first consider the network with a given coding subgraph[2] G_m for each session m. The subgraph G_m can be viewed as the union of links of a set R_m of possibly overlapping multicast trees, each connecting source s_m to all destinations $d \in D_m$. Rate control is carried out by adjusting the flow rate on each tree. Coding is done on overlapping segments of different trees of a session that have disjoint sets of downstream destinations. Figure 2 shows an example of muticast trees that are decomposed from the coding subgraph shown in Figure 1. In this example, coding on the shared link is possible, allowing both trees to simultaneously transmit information at their maximum individual rates.
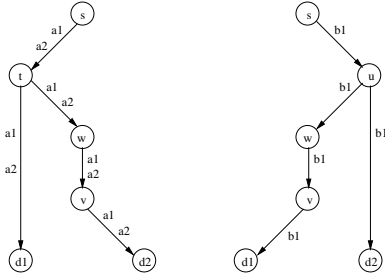

Fig. 2. Multicast trees for the example shown in Fig.1. Coding is done on the shared link (w, v), which, as part of the left tree, has one downstream destination d_2, and, as part of the right tree, has one downstream destination d_1. The left tree can support up to two units of information flow and the right tree can independently support up to one unit of information flow, since coding on link (w, v) allows the two trees to share capacity.

Analogous to practical routing, such coding subgraphs can be chosen in a variety of ways based on combinations of different considerations, such as delay, resource usage or commercial relationships among network providers. For instance, we can use existing multicast tree construction algorithms, or use existing techniques for finding multiple paths to each destination and combine appropriate sets of paths that form trees.

To simplify notation, we consider the case where overlapping segments of different trees of a session have disjoint sets of downstream destinations, thus allowing coding to occur on

---

[2]In this and the following sections, subgraph refers to "un-capacitated" subgraph.

---

all overlapping segments[3]; the more general case where coding occurs only on some overlapping segments admits a similar analysis. Each tree T_r^m, $r \in R_m$ contains a set $L_r \subset L$ of links, which defines a $|L| \times |R_m|$ multicast matrix H^m whose (l, r)th entry is given by

$$H_{lr}^m = \begin{cases} 1 & \text{if } l \in L_r \\ 0 & \text{otherwise.} \end{cases}$$

Note that over each multicast tree T_r^m the source sends the same information flow to each destination; we denote its rate by x_r^m. With intra-session network coding, the physical flow rate y_l^m for each multicast session m though link l is $y_l^m = \max_r \{H_{lr}^m x_r^m\}$. The link capacity constraints (4) become

$$\sum_m y_l^m = \sum_m \max_r \{H_{lr}^m x_r^m\} \leq c_l \ \forall l \in L. \qquad (5)$$

By Theorem 1, conditions (5) are satisfied if and only if there exists a corresponding multicast network code of rate arbitrarily close to $\sum_r x_r^m$ from source s_m to destinations $d \in D_m$.

Following [12], assume each session m attains a utility $U_m(x^m)$ when it transmits at a rate $x^m = \sum_r x_r^m$ packets per second over the coding subgraph. We assume $U_m(\cdot)$ is continuously differentiable, increasing, and strictly concave for the flows with elastic rate demand. Our objective is to choose source rates x_r^m so as to solve the following global problem

$$\mathbf{P1}: \quad \max_{x_r^m, y_l^m} \quad \sum_m U_m(x^m)$$
$$\text{subject to} \quad H_{lr}^m x_r^m \leq y_l^m, \ \forall r \in R_m, \ \forall m \in M$$
$$\sum_m y_l^m \leq c_l, \ \forall l \in L.$$

C. Multicast without Given Coding Subgraphs

Since coding subgraphs are not given, we directly use the network coding flow constraints (1)-(4) and Theorem 1, given in subsection III-A, to formulate the following optimization problem which chooses source rates x^m, information rates $g_{i,j}^{md}$ and physical flow rates $f_{i,j}^m$ so as to maximize aggregate utility:

$$\mathbf{P2}: \quad \max_{x,g,f} \quad \sum_m U_m(x^m)$$
$$\text{subject to} \sum_{j:(i,j)\in L} g_{i,j}^{md} - \sum_{j:(j,i)\in L} g_{j,i}^{md} = x_i^m, i \neq d, \forall d, m$$
$$g_{i,j}^{md} \leq f_{i,j}^m, \ \forall d, m$$
$$\sum_m f_{i,j}^m \leq c_{i,j}, \ \forall(i,j) \in L,$$

where $x_i^m = x^m$ if $i = s_m$ and $x_i^m = 0$ otherwise. Here we do not include flow balance equation at destinations, which is automatically guaranteed by the flow balance at the source and intermediate nodes.

Note that in the models **P1** and **P2**, network coding comes into action through the constraints (5) and (3). With Theorem 1, this gives some form of "separation principle" that

---

[3]This is the case, for instance, if each session's trees have been formed by first finding multiple link-disjoint paths to each destination and then choosing combinations of these paths that form trees.

allows us to separate decisions on resource usage and rate control from the design of the actual network codes.

The system problems **P1** and **P2** are convex optimization problems, and are polynomially solvable if all the utilities and constraint information is provided, but this is impractical in real networks. Since they are convex optimization problems with strong duality, distributed algorithms can be derived by formulating and solving corresponding Lagrange dual problems, as we will show in the next two sections.

IV. DISTRIBUTED ALGORITHM FOR NETWORKS WITH GIVEN CODING SUBGRAPHS

We introduce for each multicast session m traffic split variables $\alpha_r^m \geq 0$ for each multicast tree T_r^m of the coding subgraph, such that $\sum_r \alpha_r^m = 1$ and $x_r^m = x^m \alpha_r^m$. We see that α_r^m controls the fraction of the traffic of multicast session m that is sent through the tree T_r^m. Instead of solving the problem **P1** directly, we first consider the version of the rate control problem with the fixed split vector α.

$$\textbf{P1a}: \quad \max_{\{x^m, y_l^m\}} \quad \sum_m U_m(x^m)$$
$$\text{subject to} \quad H_{lr}^m x^m \alpha_r^m \leq y_l^m$$
$$\sum_m y_l^m \leq c_l.$$

The above problem is a strictly convex and has a unique solution, with respect to source rates x^m. Let us denote its maximum by $U(\alpha)$. The system problem **P1** corresponds to computing

$$\textbf{P1b}: \quad \max_{\alpha \geq 0} \quad U(\alpha)$$
$$\text{subject to} \quad \sum_r \alpha_r^m = 1.$$

Note that the above problem is not necessarily convex. But we will see later that it can still be solved for globally optimality.

A. Two-Timescale Distributed Algorithm

Consider the Lagrangian of the problem **P1a** with respect to the constraints due to network coding

$$L(\alpha, p, x, y) = \sum_m U_m(x^m) - \sum_{l,m,r} p_{l,r}^m (H_{lr}^m x^m \alpha_r^m - y_l^m).$$

Interpreting $p_{l,r}^m$ as the "congestion price" at link l for multicast tree T_r^m and maximizing the Lagrangian over x and y for fixed p, we obtain the following joint rate control and session scheduling algorithm:

Rate control: At time t, given congestion price $p(t)$, the source s_m adjusts flow rate x^m according to the aggregate congestion price $\sum_l H_{l,r}^m p_{l,r}^m$ over the multicast trees T_r^m,

$$x^m(t) = (U_m')^{-1}(\sum_r \alpha_r^m \sum_l H_{l,r}^m p_{l,r}^m(t)). \qquad (6)$$

Similar to TCP congestion control algorithm where the source adjusts its sending rate according to aggregate congestion price along its path, this rate control mechanism has the desired price structure and is an end-to-end congestion control mechanism.

Session scheduling: Random linear network coding scheme is used to code packets from the same multicast session [9]. Over link l, send an amount of coded packets for the session[4] $m_l(t) = \arg\max_m \sum_r p_{l,r}^m(t)$ at rate c_l. Mathematically, this is because the maximization over $\{y_l^m\}$ is a linear programming and we can always choose an extreme point solution. This is equivalent to maximizing over y by the following assignment

$$y_l^m(t) = \begin{cases} c_l & \text{if } m = m_l(t) \\ 0 & \text{if } m \neq m_l(t). \end{cases} \qquad (7)$$

Defining $D(\alpha, p) = \max_{x,y} L(\alpha, p, x, y)$ with $\sum_m y_l^m \leq c_l$, by duality we have (see, e.g., Chapter 5 in [2])

$$U(\alpha) = \min_{p \geq 0} D(\alpha, p) = \min_{p \geq 0} \max_{x,y} L(\alpha, p, x, y).$$

The dual problem $\min_p D(\alpha, p)$ can be solved by using the subgradient method [24], [2], where the Lagrangian multipliers are adjusted in the opposite direction to the subgradient $\partial_p D(\alpha, p)$.

Congestion price update: Each link l updates its price with respect to multicast tree T_r^m, according to

$$p_{l,r}^m(t+1) = [p_{l,r}^m(t) + \gamma_t (H_{lr}^m \alpha_r^m x^m(p(t)) - y_l^m(p(t)))]^+, \quad (8)$$

where γ_t is a positive scalar stepsize, and '+' denotes the projection onto the set $\Re^+$ of non-negative real numbers. Note that link l will use capacity y_l^m to transfer coded packets for multicast session m, equation (8) says that if the demand $H_{lr}^m x_r^m$ for virtual capacity at link l for the information flow of multicast tree T_r^m exceeds the assigned physical capacity y_l^m, the price p_{lr}^m will rise, and decreases otherwise. Also, note that equation (8) is distributed and can be implemented at individual links using only local information.

The above rate control algorithm (6)-(8) works under the assumption that the traffic split vector α remains constant. We now discuss how to adjust α_r^m to solve the problem **P1b**, which we call tree adaptation. We assume that tree adaptation is much slower so that the minimization of $D(\alpha, p)$ over p can be seen as instantaneous.

Intuitively, the optimal traffic split vector should strike an equilibrium that is similar to Wardrop equilibrium, where for each multicast session the aggregate prices in all multicast trees actually used are equal and less than those which would be experienced by a single packets on any unused tree [26]. We gradually update the split vector towards this equilibrium, as in [6], [22]. At stage n, given split vector $\alpha(n)$, suppose $p(n) \in \arg\min_p D(\alpha(n), p)$ is an optimum solution to the dual problem of **P1a**, and let $r^m(n) = \arg\min_r \sum_l p_{l,r}^m(n)$.

Tree adaptation: Each source s_m updates α_r^m according to

$$\alpha_r^m(n+1) = \alpha_r^m(n) + \delta_r^m(n), \qquad (9)$$

with

$$\delta_r^m(n) = \begin{cases} -\min\{\alpha_r^m(n), \kappa_n(\sum_l p_{l,r}^m(n) - \sum_l p_{l,r^m(n)}^m(n))\}, \\ \qquad \text{if } r \neq r^m(n), \\ -\sum_{r \neq r^m(n)} \delta_r^m(n), \\ \qquad \text{if } r = r^m(n), \end{cases} \qquad (10)$$

---

[4]When there exists more one session with maximal aggregate congestion price, the tie is broken randomly.

where κ_n is a positive scalar stepsize. It is straightforward to verify that

$$\sum_r \delta_r^m(n) = 0, \qquad (11)$$

$$\sum_r \delta_r^m(n) \sum_l p_{l,r}^m \leq 0. \qquad (12)$$

We see that $\sum_r \delta_r^m(n) \sum_l p_{l,r}^m = 0$ only if $\delta_r^m(n) = 0$, which requires

$$\alpha_r^m(n)(\sum_l p_{l,r}^m(n) - \sum_l p_{l,r^m(n)}^m(n)) = 0. \qquad (13)$$

This is exactly the Wardrop equilibrium. Also, note that equations (11)-(13) are similar to the condition specified for the route adaptation in [6], [22].

B. Convergence Analysis

The distributed rate control algorithm presented in last subsection has embedded loops. In the inner loop (6)-(8), which operates at a fast timescale, the network searches for optimal source rates and congestion prices for fixed flow split vector. In the outer loop (9)-(10), which operates at a slow, traffic engineering timescale, the sources adapt the flow split vector based on the stabilized congestion prices in the network. The tree adaptation algorithm (9)-(10) can be seen as a method for stable traffic engineering based on congestion prices.

We now provide the convergence analysis of the inner loop algorithm (6)-(8). Denote P^* as the set of optimal solutions to the dual problem $\min_p D(\alpha, p)$. Define $\rho(p, S) = \min_{\bar{p} \in S} ||p - \bar{p}||$ as the Euclidean distance of a point p to set S. Directly applying the convergence results for the subgradient method [24], we have the following theorem.

Theorem 2: If the stepsizes γ_t satisfy the following condition

$$\lim_{t \to \infty} \gamma_t = 0, \quad \sum_{t=0}^{\infty} \gamma_t = \infty, \qquad (14)$$

then the iterative algorithm (8) converges, i.e.,

$$\lim_{t \to \infty} \rho(p(t), P^*) = 0. \qquad (15)$$

Note that x^m is a continuous function of the congestion price $p_{l,r}^m$. Let x^* be the optimal solution of the problem **P1a**. By duality, the following result is immediate.

Corollary 3: Under the same condition as in (14), the iterative algorithm (6)-(8) converges, i.e.,

$$\lim_{t \to \infty} ||x(t) - x^*|| = 0. \qquad (16)$$

In practical implementation, a constant stepsize rather than the diminishing stepsize as in (14) is desired. For a constant stepsize, the subgradient method may not converge to an optimal solution, but can be made to converge within any given small neighborhood around the optimum, by choose sufficiently small constant stepsize (see, e.g., [24] for details). Since the source rates are continuous function of the congestion price, the optimality of the algorithm (6)-(8) is guaranteed with arbitrary precision by appropriately choosing a constant stepsize.

Now we study the convergence of the outer loop algorithm. For simplicity of the presentation, we consider the continuous time version of the algorithm (9)-(10), which satisfies

$$\sum_r \dot{\alpha}_r^m = 0, \qquad (17)$$

$$\sum_r \dot{\alpha}_r^m \sum_l p_{l,r}^m \leq 0. \qquad (18)$$

Similarly, we have $\sum_r \dot{\alpha}_r^m \sum_l p_{l,r}^m = 0$ only if $\dot{\alpha}_r^m = 0$, which happens only if (13) is satisfied. The tree adaptation algorithm (9)-(10) can be seen as a specific discrete time implementation of the adaptation algorithm (17)-(18).

Theorem 4: The tree adaptation algorithm (17)-(18) converges to the optimal solution of the system problem **P1**.

Proof: Note that

$$U(\alpha) = \min_p D(\alpha, p)$$
$$= \min_p \{\sum_m U_m(x^m(p))$$
$$- \sum_{m,l,r} p_{l,r}^m (H_{l,r}^m \alpha_r^m x^m(p) - y_l^m(p))\}, \quad (19)$$

and $D(\alpha, p)$ as a function of p includes a smooth term plus a non-smooth piece-wise linear term. So, the differential of $U(\alpha)$ can be written as

$$dU(\alpha) = \left(\lim_{h \to 0^+} \frac{\partial D(\alpha, p^* + hdp)}{\partial p} \right) dp + \frac{\partial D(\alpha, p^*)}{\partial \alpha} d\alpha$$
$$= \left(\lim_{h \to 0^+} \frac{\partial D(\alpha, p^* + hdp)}{\partial p} \right) dp - \sum_{m,l,r} x^m H_{l,r}^m (p^*)_{l,r}^m d\alpha_r^m, \quad (20)$$

where $p^* = \arg\min_p D(\alpha, p)$. Since p^* minimizes $D(\alpha, p)$ given α, $\lim_{h \to 0^+} \frac{\partial D(\alpha, p^* + hdp)}{\partial p}$ cannot be a descent direction. So, $\left(\lim_{h \to 0^+} \frac{\partial D(\alpha, p^* + hdp)}{\partial p} \right) dp \geq 0$. Hence,

$$dU(\alpha) \geq - \sum_{m,l,r} x^m H_{l,r}^m (p^*)_{l,r}^m d\alpha_r^m, \qquad (21)$$

i.e.,

$$\dot{U}(\alpha) \geq - \sum_{m,l,r} x^m H_{l,r}^m (p^*)_{l,r}^m \dot{\alpha}_r^m. \qquad (22)$$

By (17)-(18), we have $\dot{U}(\alpha) \geq 0$. So, the tree adaptation algorithm (17)-(18) will converge to an equilibrium α^* such that $\dot{U}(\alpha^*) = 0$. However, this only guarantees the convergence of the tree adaptation algorithm. Without further elaboration, we cannot even claim it solves a local optimal of the problem **P1b**.

Note that, following equations (6) and (13), we obtain at $(\alpha^*, p(\alpha^*), x(\alpha^*))$

$$U_m'(x^m) = \frac{\partial U_m}{\partial x_r^m}(x^m) = \sum_l H_{l,r}^m p_{l,r}^m, \text{ if } x_r^m > 0, \quad (23)$$

$$\sum_l H_{l,r}^m p_{l,r}^m > U_m'(x^m), \text{ if } x_r^m = 0, \quad (24)$$

which means that

$$x(\alpha^*) = \arg\max_{x_r^m} \sum_m U_m(\sum_r x_r^m) - \sum_{m,r,l} p_{l,r}^m(\alpha^*) H_{l,r}^m x_r^m. \quad (25)$$

Also, we have $y(\alpha^*) = \arg\max_y \sum_{m,r,l} p_{l,r}^m(\alpha^*) y_l^m$. Denote the Lagrangian of the system problem **P1** with respect to the constraints due to network coding as $\hat{L}(p, x, y)$. We have

$$(x(\alpha^*), y(\alpha^*)) = \arg\max_{x,y} \hat{L}(p(\alpha^*), x, y). \qquad (26)$$

Furthermore, by duality between the problem **P1a** and its dual, we have

$$\sum_{m,l,r} p_{l,r}^m(\alpha^*)(H_{l,r}^m x_r^m(\alpha^*) - y_l^m(\alpha^*)) = 0. \qquad (27)$$

Combining (26)-(27), we conclude that

$$\sum_m U_m(\sum_r x_r^m(\alpha^*)) = \hat{L}(p(\alpha^*), x(\alpha^*), y(\alpha^*)), \qquad (28)$$

which by duality only happens when $p(\alpha^*)$ and $x_r^m(\alpha^*)$ solve the system problem **P1** and its dual. So, the tree adaptation algorithm (17)-(18) indeed solves the system problem **P1**. This also proves that the tree adaptation algorithm solves the problem **P1b**. ∎

Equations (20)-(21) can be seen as a generalized Envelope Theorem for nonsmooth functions. If $D(\alpha, p)$ is a smooth function of p, we have $\lim_{h\to 0^+} \frac{\partial D(\alpha, p^* + h dp)}{\partial p} = \frac{\partial D(\alpha, p^*)}{\partial p} = 0$ along any direction dp, and thus recover usual Envelope theorem, see [20]. Also, note that the adaptation algorithm (9)-(10) and Theorem 4 can be readily extended to routing-based multicasting and multipath routing.

C. Implementation of Price Feedback

Each link l keeps a separate virtual queue p_{lr}^m for each multicast tree T_r^m of each session m which acts as the congestion price. Each packet's header contains the indexes of the trees whose information it contains. When a packet is received at a node from an incoming link l, if the packet header contains the rth tree index, the queue size p_{lr}^m is increased by one; otherwise it is unchanged. Similarly, when a packet is sent by a node on an outgoing link l, if the packet header contains the rth tree index, the queue size p_{lr}^m is decreased by one; otherwise it is unchanged. The congestion prices over a multicast tree are fed back to the source node in the following way. Each node i in the tree will pass the aggregate price along the links from the receivers till itself to the upstream node j ("upstream" is defined as the direction from receivers to source node over a multicast tree). In this recursive way, the source node will get the aggregate congestion prices over that multicast tree, and adjust the sending rate accordingly.

V. DISTRIBUTED ALGORITHM FOR NETWORKS WITHOUT GIVEN CODING SUBGRAPHS

A. Distributed Algorithm

Now we turn to system problem **P2** and consider its Lagrangian with respect to the flow balance constraints,

$$\begin{aligned} L(p, x, g, f) &= \sum_m U_m(x^m) - \sum_{i,m,d\in D_m} p_i^{md}(x_i^m \\ &\quad - \sum_{j:(i,j)\in L} g_{i,j}^{md} + \sum_{j:(j,i)\in L} g_{j,i}^{md}). \end{aligned}$$

Interpreting p_i^{md} as the "congestion price" at node i for multicast session m and destination $d \in D_m$ and maximizing the Lagrangian over x, g and f for fixed p, we obtain the following joint rate control and session scheduling algorithm:

Rate control: At time t, given congestion price $p(t)$, each source node s_m adjusts its sending rate according to local congestion price that is generated locally at the source node,

$$x_m(t) = {U_m'}^{-1}(\sum_{d\in D_m} p_{s_m}^{md}(t)). \qquad (29)$$

Note that

$$\begin{aligned} &\max_{g,f} \sum_{i,m,d} p_i^{md}(\sum_j g_{i,j}^{md} - \sum_j g_{j,i}^{md}) \quad s.t. \quad g_{i,j}^{md} \le f_{i,j}^m \\ =\ &\max_{g,f} \sum_{i,j,m,d} g_{i,j}^{md}(p_i^{md} - p_j^{md}) \quad s.t. \quad g_{i,j}^{md} \le f_{i,j}^m \\ =\ &\max_f \sum_{i,j,m,d} f_{i,j}^m[p_i^{md} - p_j^{md}]. \end{aligned}$$

Each node i collects congestion price information from its neighbor j, find multicast session $m_{i,j}(t)$ such that

$$m_{i,j}(t) \in \arg\max_m \sum_{d\in D_m} [p_i^{md}(t) - p_j^{md}(t)]^+,$$

and calculate differential price $w_{i,j}(t) = \sum_d [p_i^{m_{i,j}(t)d}(t) - p_j^{m_{i,j}(t)d}(t)]^+$.

Session scheduling: Over link (i,j), a random linear combination of data of multicast session $m_{i,j}$ to all destinations d such that $p_i^{m_{i,j}d}(t) - p_j^{m_{i,j}d}(t) > 0$ is sent at rate c_l. Mathematically, this is equivalent to solving the primal variable g by the following assignment

$$g_{i,j}^{md}(t) = \begin{cases} c_{i,j} & \text{if } m = m_{i,j}(t) \ \& \ p_i^{md}(t) - p_j^{md}(t) > 0 \\ 0 & \text{otherwise.} \end{cases} \qquad (30)$$

Define

$$\begin{aligned} D(p) &= \max_{x_m, g_{i,j}^{md}, f_{i,j}^m} \quad L(p, x, g, f) \\ &\text{subject to} \quad g_{i,j}^{md} \le f_{i,j}^m, \ \sum_m f_{i,j}^m \le c_l. \end{aligned}$$

Again the dual problem $\min_p D(p)$ can be solved by using the subgradient method.

Congestion price update: Each node i updates its price with respect to multicast session m and destination $d \in D_m$, according to

$$\begin{aligned} p_i^{md}(t+1) &= [p_i^{md}(t) + \gamma_t (\ x_i^{md}(p(t)) & (31) \\ &\quad - \sum_{j:(i,j)\in L} g_{i,j}^{md}(p(t)) + \sum_{j:(j,i)\in L} g_{j,i}^{md}(p(t))\ )]^+, \end{aligned}$$

and passes the price p_i^{md} to all its neighbors.

With our algorithm, each source node adjusts its sending rate according to the local congestion price. Thus, there is no communication overhead for rate control. The majority of communication overhead is for session scheduling, but that only requires nodes to communicate with direct neighbors. Thus, our design has very low communication overhead, compared with other schemes with similar models [19], [30], [28], [29]. Note that the above session scheduling component

uses *back-pressure* to do optimal scheduling, similarly to [10]. Such dynamic network coding based multicasting offers both a larger rate region and much lower complexity, as compared to optimal dynamic routing based multicasting [23].

B. Convergence Analysis

The algorithm (29)-(31) is a subgradient algorithm. Thus, Theorems 2 and 3 and related discussion apply. Using the same notation as in subsection IV-B, we have the following result regarding the convergence property of the joint rate control and session scheduling algorithm for multicast without given coding subgraph.

Theorem 5: If the stepsizes γ_t satisfy the following condition

$$\lim_{t \to \infty} \gamma_t = 0, \quad \sum_{t=0}^{\infty} \gamma_t = \infty,$$

then the iterative algorithm (29)-(31) converges, i.e.,

$$\lim_{t \to \infty} \rho(p(t), P^*) = 0,$$
$$\lim_{t \to \infty} ||x(t) - x^*|| = 0.$$

C. Implementation of Price Feedback

Since the scheme is destination-based, each packet need to carry a vector of destination identities in the packet header, in addition to coding vector. Each nodes i keep a separate virtual queue p_i^{md} as congestion price for each multicast session m and destination $d \in D_m$. The arrival and the departure of these queues evolve as follows. When a packet is received at node i, i will check the destination vector in the header of this packet. If this packet is intended for destination d , the queue size p_i^{md} will increase by one; Otherwise, the virtual queue size will remain the same. When a packet is sent out at node i, i will check the destination vector of this packet. If this packet is intended for destination d , the queue size p_i^{md} will decrease by one; Otherwise, the virtual queue size will remain the same. Note that, here we use back-pressure to do rate control. The source nodes s adjust the sending rate according to local congestion prices at s, and the congestion in the network is propagated to the source node through back-pressure.

VI. NUMERICAL EXAMPLES

In this section, we provide numerical examples to complement the analysis in previous sections. We consider a simple network shown in the left graph in Figure 3. The network is assumed to be undirected and each link has equal capacities in both directions. Assume that there are two multicast sessions, session one with source node s and destinations x and y and session two with source node t and destination u and z, with the same utility $U_m(x_m) = \log(x_m)$. We have chosen such a small, simple topology to facilitate detailed discussion of the results.
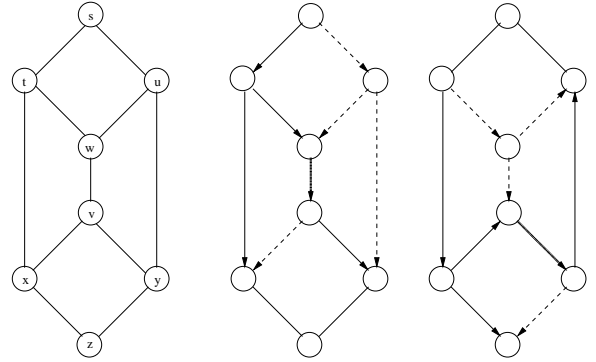

Fig. 3. A simple network with two multicast sessions. The given coding subgraphs for sessions 1 and 2 are shown in the middle and right graphs respectively. For each session, the first tree is indicated by solid arrows, the second by dashed arrows, and the overlapping segments by bold arrows.

A. Muticasting with Given Coding Subgraphs

We assume that the given coding subgraphs for sessions one and two are those shown in the middle and right graphs of Figure 3 respectively. The subgraph for each session decomposes into two multicast trees in the same way as in Figure 2. For simplicity, we assume the following link capacities: link (s, t) has 2 units of capacity, links (t, x) and (v, y) have 5 units of capacity, links (s, u), (u, w) and (y, z) have 1 unit of capacity and all other links have 3 units of capacity.

Figure 4 shows the evolution of source rates (left panel) versus the number of iterations of the outer loop tree adaptation algorithm and the evaluation of traffic split vectors (middle panel) with stepsize $\kappa_n = 0.01$. It can be seen from the plots that the source rates are well within 5% of their optimal values after 10 iterations, and the traffic split vectors are well within 5% of their optimal values after 15 iterations. The simulation result also shows that for session one, the traffic is 2 over the first tree and 1 over the second tree, and for session two, the traffic is 3 over the first tree and 1 over the second tree. In this simulation, the inner loop congestion control algorithm runs 500 iterations before each run of the tree adaptation algorithm. Comparable performance is observed even if the number of inner loop iterations is as low as 100. So, the convergence of the whole rate control algorithm is very fast.

In practice, the end users can dynamically control the number of iterations, by monitoring the congestion prices over different multicast trees. The right panel of Figure 4 shows the evolution of the congestion prices over different trees versus the number of iterations of the tree adaptation algorithm. We can, for instance, specify a threshold value and decide the whole algorithm has converged when the relative differences in price over different multicast trees are less than the threshold value. The users can also set the stepsize of the tree adaptation algorithm dynamically. When the price differences over different trees are large, the user can choose a large stepsize, and when the differences are small, he can choose a small stepsize.

B. Muticasting without Given Coding Subgraphs

We now consider the same network but without given coding subgraphs. The distributed algorithm developed in section
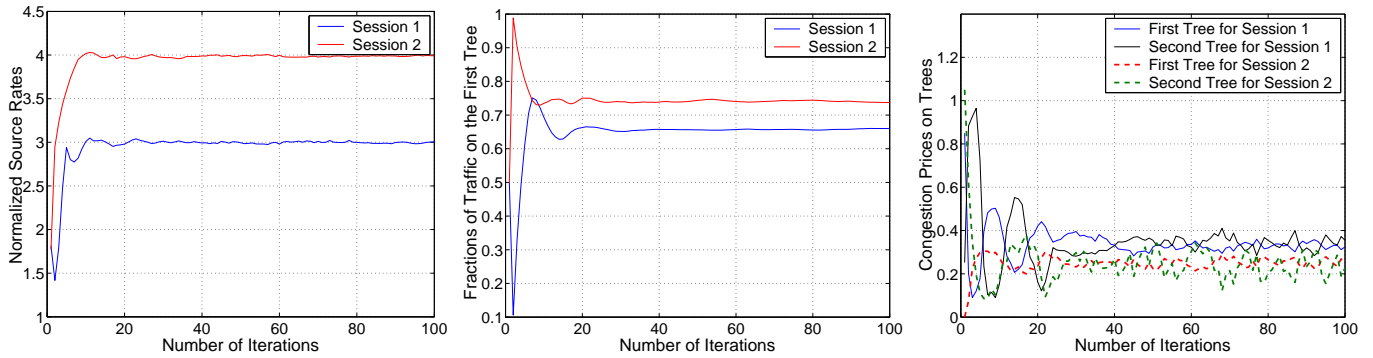
Fig. 4. The evolution of source rates (left panel), the evolution of traffic split vectors (middle panel), and the evolution of congestion prices over different multicast trees (right panel) versus the number of iterations of the tree adaptation algorithm with stepsize $\kappa_n = 0.01$ for the example network with given coding subgraphs.

V will go through the whole network (the undirected graph on the left side in Figure 3) to find capacitated coding subgraphs that maximize the aggregate utility. For this example, we assume the following link capacities: links (s,t), (t,x) and (x,v) have 2 units of capacity, links (t,w), (w,v) and (v,y) have 3 units of capacity and all other links have 1 unit of capacity.

Figure 5 shows the evolution of the source rates with the constant stepsize $\gamma_t = 0.01$. We see that the source rates approach the corresponding stable values 3 and 2 quickly. The simulation result also shows coding occurs over the same subgraphs as those in Fig.3: 2 units of traffic of session one is coded over link (w,v) and 2 units of traffic of session two is coded over link (v,y). It is not difficult to check that those are optimal source rates and coding subgraphs. Also, note that the system converges to within a small neighborhood of the optimal, since we have chosen a constant stepsize. In order to study the impact of different choices of the stepsize on the convergence of the algorithm, we have run simulations with different stepsizes. We found that the smaller the stepsize, the slower the convergence and the closer to the optimal, which is a general characteristic of any gradient based method. So, there is a tradeoff between convergence speed and optimality. In practice, the end user can first choose large stepsizes to ensure fast convergence, and subsequently, the stepsizes can be reduced once the source rate starts oscillating around some mean value.
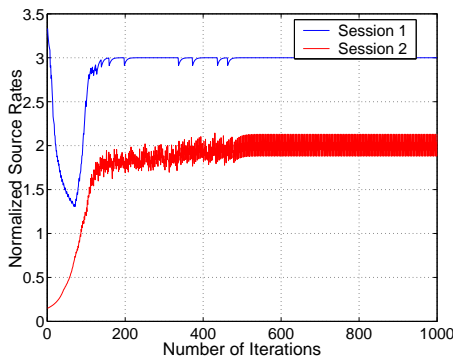
C. Comparison of the Two Algorithms

To compare the performance of the two rate control algorithms, we consider the same network, with 1 unit capacity for each link. Figure 6 shows the evolution of the source rates versus the number of iterations of the tree adaptation algorithm for the case with given coding subgraphs as shown in the right side graph of Figure 3, and the evolution of the source rates for the case without given coding subgraphs. We see that the throughput achieved for the case without given subgraphs is larger than that for the case with given coding subgraphs. This is expected, since the capacity region for the case with given coding subgraph is a subset of the capacity region with the coding subgraphs unspecified.
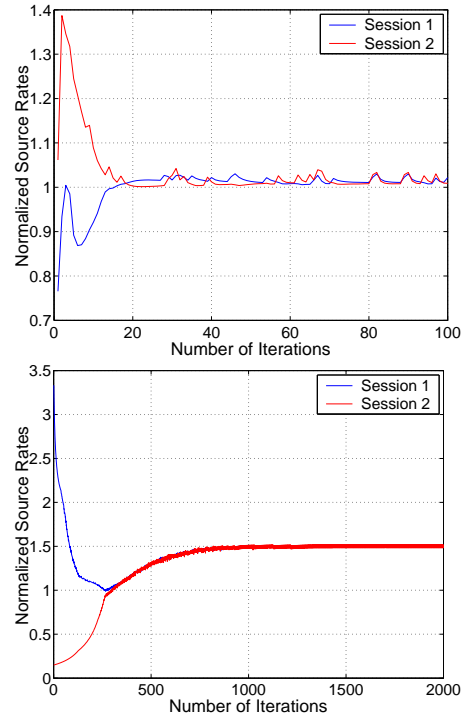

Fig. 6. The evolution of source rates for the case with given coding subgraph (upper panel) and for the case without given coding subgraph (lower panel).

Fig. 5. The evolution of source rates with step size $\gamma_t = 0.01$ for the example network without given coding subgraph.

VII. Some Further Remarks

There may exist several ways to solve the system problems **P1** and **P2**. The challenge is to find distributed solutions that respect as much as possible the information constraints of the Internet and can be implemented at the sources and routers. This requires to minimize information and respect signaling mechanism for adaptive control in Internet as much as possible. So, we choose not to relax all the constraints when solving for the duals. Besides the equilibrium, dynamics are also important in our consideration. In section IV, in order to avoid "tree" oscillation, we achieve rate control through a combination of fast timescale rate control and slow, traffic engineering timescale traffic splitting. Also, note that there is an explicit session scheduling component in our algorithms, which requires per-session information. This seems to incur much complexity in implementation. However, with network coding, network nodes need to keep tracking per-session information. We just exploit this additional complexity to the good.

The two algorithms developed in sections IV and V can coexist: some multicast sessions adopt the algorithm with given coding subgraphs and other sessions adopt the algorithm without given coding subgraphs; and they are coupled through the flow balance equations at nodes and capacity constraints at links. Also, unicasting can be seen as special case of multicasting. Mathematically, in the system model **P1**, network coding comes into action through constraint $H_{lr}^m x_r^m \leq y_l^m$, and in system model **P2**, network coding comes into action through constraint $g_{i,j}^{md} \leq f_{i,j}^m$. It is straightforward to include uncoded unicast flows into the system models and carry out these iterative algorithms in the same way, with only slightly more complicated notation.

VIII. Conclusions

We have presented two models for rate control for multicast with network coding, one for networks with given coding subgraphs, and one where such subgraphs are found dynamically. We developed two corresponding dual-based adaptive rate control algorithms. With random network coding, both algorithms can be implemented in a distributed manner, and work at transport layer to adjust source rates and at network layer to do network coding. We prove that the proposed algorithms converge to the globally optimal solutions for each model. Numerical examples are provided to complement our theoretical analysis. We will further study the practical implementation of our algorithms. Also, how to obtain optimal coding subgraphs based on general cost criteria is an interesting problem. Solving this problem will further facilitate the practical deployment of network coding in real networks.

References

[1] R. Ahlswede, N. Cai, S.-Y. R. Li and R. W. Yeung, Network information flow, *IEEE Trans. on Information Theory*, 46:1204-1216, 2000.

[2] D. Bertsekas, *Nonlinear Programming*, 2nd ed., Athena scientific, 1999.

[3] P. A. Chou, Y. Wu and K. Jain, Practical network coding, *Proc. Allerton Conference on Communication, Control and Computing*, 2003.

[4] A. F. Dana, R. Gowaikar, R. Palanki, B. Hassibi and M. Effros, Capacity of wireless erasure networks, *IEEE Trans. on Information Theory*, 2006.

[5] S. Deb and R. Srikant, Congestion control for fair resource allocation in networks with multicast flows, *Proc. CDC*, 2001.

[6] R. Gallager, A minimum delay routing algorithm using distributed computation, *IEEE Transactions on Communication*, 25(1):73-85, 1977.

[7] T. Ho, R. Koetter, M. Medard, D. R. Karger and M. Effros, The Benefits of Coding over Routing in a Randomized Setting, *Proc. of IEEE International Symposium on Information Theory*, 2003.

[8] T. Ho M. Medard, R. Koetter, An Information Theoretic View of Network Management, *IEEE Trans. on Information Theory*, 51:1295-1312, 2005.

[9] T. Ho, M. Medard, R. Koetter, D. Karger, M. Effros, J. Shi, and B. Leong, A Random Linear Network Coding Approach to Multicast, *IEEE Transactions on Information Theory*, 52(10):4413–4430, October 2004.

[10] T. Ho and H. Viswanathan, Dynamic algorithms for multicast with intra-session network coding, *Proc. Allerton Conference on Communication, Control and Computing*, 2005.

[11] K. Kar, S. Sarkar and L. Tassiulas, Optimization based rate control for multirate multicast sessions, *Proc. IEEE Infocom*, 2001.

[12] F. P. Kelly, A. K. Maulloo and D. K. H. Tan, Rate control for communication networks: Shadow prices, proportional fairness and stability, *Journal of Operations Research Society*, 49(3):237-252, March 1998.

[13] R. Koetter, M. Medard, An Algebraic Approach to Network Coding, *IEEE/ACM Transactions on Networking*, 11:782-795, 2003.

[14] S. Kunniyur and R. Srikant, End-to-end congestion control schemes: Utility functions, random losses and ECN marks, *IEEE/ACM Transactions on networking*, 11(5):689-702, October 2003.

[15] S.-Y. R. Li, R. W. Yeung, and N. Cai, Linear network coding, *IEEE Transactions on Information Theory*, 49:371-381, 2003.

[16] X. Lin, N. Shroff and R. Srikant, A tutorial on cross-layer otimization in wireless networks, *IEEE Journal on Selected Areas in Communications*, 24(8):1452-1463, 2006.

[17] S. H. Low and D. E. Lapsley, Optimal flow control, I: Basic algorithm and convergence, *IEEE/ACM Trans. on networking*, 7(6):861-874, 1999.

[18] D. S. Lun, M. Medard and M. Effros, On coding for reliable communication over packet networks, *Proc. of Allerton Conference on Communication, Control, and Computing*, 2004.

[19] D. S. Lun, N. Ratnakar, M. Medard, R. Koetter, D. R. Karger, T. Ho and E. Ahmed, Minimum-cost multicast over coded packet networks, *IEEE Trans. Inform. Theory*, 2006.

[20] A. Mas-Colell, M. Whinston and J. Green, *Microeconomic Theory*, Oxford University Press, 1995.

[21] M. Neely, E. Modiano and C. Rohrs, Dynamic power allocation and routing for time varying wireless networks *Proc. IEEE Infocom*, 2003. Journal version, *IEEE J. Sel. Area Comm.*, 23(1):89-103, 2005.

[22] F. Pagnini, Congestion control with adaptive multipath routing based on optimization, *Proc. CISS*, 2006.

[23] Saswati Sarkar and Leandros Tassiulas. A framework for routing and congestion control for multicast information flows. *IEEE Transactions on Information Theory*, 2002.

[24] N. Z. Shor, *Minimization Methods for Non-Differentiable Functions*, Springer-Verlag, 1985.

[25] L. Tassiulas and A. F. Ephremides, Stability Properties of Constrained Queuing Systems and Scheduling Policies for Maximum Throughput in Multihop Networks, *IEEE Transactions on Information Theory*, 1992.

[26] J. G. Wardrop, Some theoretical aspects of road traffic research, Proceedings, Institute of Civil Engineers, PART II, Vol.1, pp. 325-378.

[27] Y. Wu, P. A. Chou, Q. Zhang, K. Jain, W. Zhu and S. Y. Kung, Network Planning in Wireless Ad Hoc Networks: A Cross-Layer Approach, *IEEE Journal on Selected Areas in Communications*, 2005.

[28] Y. Wu, M. Chiang and S.Y. Kung, Distributed utility maximization for network coding based multicasting: A critical cut approach, *Proc. IEEE NetCod*, 2006.

[29] Y. Wu and S.Y. Kung, Distributed utility maximization for network coding based multicasting: A shortest path approach, *IEEE Journal on Selected Areas in Communications*, 2006.

[30] Y. Xi and E. M. Yeh, Distributed algorithms for minimum cost multicast with network coding, *Proc. Allerton Conference*, 2005.

[31] R. W. Yeung Multilevel diversity coding with distortion, *IEEE Trans. on Information Theory*, 1995.